

1. <https://final-project-314709.uc.r.appspot.com>

2. <https://final-project-314709.uc.r.appspot.com>

3. This project is using Auth0 to be the JWT provider, and Google Cloud Datastore to be the data store. There are 3 entities: User, Boat, Load.

A user can have any number of boats, but a boat can have only one owner. A boat can have multiple loads on the boat, but a load can have up to one carrier and the load can have no carrier (carrier is empty). Every boat can only be created by the user, and once a boat is created, it's owner is the creator. This relationship can not be modified later.

In any endpoints, the id field will only be the id that Google datastore uses. For example, a GET request send to the endpoint "host_url/boats/boat_id", the boat_id can only be the 'id' property of the user, if you use 'user_id' property of the user, the response from the server will be 404 Not Found.

Before using postman collections, go to login page and paste & copy JWTs for two users to the postman environment.

There are two accounts that are available (you can create your own, too):

Username: neo@happy.com

Password: lin@1998

Username: foo@bar.com

Password: lin@1998

4. Data models:

A. User

```
{  
  id: 12321321313, (int, auto generated)  
  user_id: "auth0|60djsnaebakjsnedlahfjn" (string, auto generated)  
}
```

'id' property indicates the id that Google datastore used.

'user_id' property indicates the id that Auth0 uses to generate JWT.

B. Boat

```
{  
  type: "a", (string, required)  
  creation_date: "2/10/2019", (string, required)  
  name: "neo" (string, required, unique)  
}
```

C. Load

```
{  
  weight: "1000kg", (string, required)  
  creation_date: "2/10/2010", (string, required)  
  length: "10m" (string, required)  
}
```

5. API specification

Endpoint: POST /boats

Protected: Yes

Possible return status codes: 201, 401, 403, 406

Sample request for 201:

Header: Authorization = Bearer {{jwt1}}

Body:

```
{
  "type": "a",
  "creation_date": "2/10/2019",
  "name": "neo"}
```

Sample Response for 201:

Body:

```
{
  "owner": 5634161670881280,
  "type": "a",
  "creation_date": "2/10/2019",
  "name": "neo",
  "loads": null,
  "self": "https://final-project-
314709.uc.r.appspot.com/boats/5681556802764800",
  "id": 5681556802764800
}
```

Sample request for 401:

Header: Authorization = empty

Body:

```
{
  "type": "a",
  "creation_date": "2/10/2019",
  "name": "neo"
}
```

Sample request for 403:

Header: Authorization = Bearer {{jwt1}}

Body:

```
{
  "type": "a",
  "creation_date": "2/10/2019",
  "name": "neo" (name is duplicated to other boats)
}
```

Sample request for 406:

Header: Authorization = Bearer {{jwt1}}

Accept_mimetypes: text/html

Body:

```
{
  "type": "a",
  "creation_date": "2/10/2019",
```

```
        "name": "neo"
    }

```

Endpoint: GET /boats

Protected: Yes

Possible return status codes: 200, 401, 406

Sample request for 200:

Header: Authorization = Bearer {{jwt1}}

Body:

```
{
    "type": "a",
    "creation_date": "2/10/2019",
    "name": "neo"
}

```

Sample request for 401:

Header: Authorization = empty

Body:

```
{
    "type": "a",
    "creation_date": "2/10/2019",
    "name": "neo"
}

```

Sample request for 406:

Header: Authorization = empty

Accept_mimetypes: text/html

Body:

```
{
    "type": "a",
    "creation_date": "2/10/2019",
    "name": "neo"
}

```

Endpoint: DELETE /boats

Protected: Yes

Possible return status codes: 405

Sample Response for 405:

```
{
    "Reject request"
}

```

Endpoint: GET /loads

Protected: No

Possible return status codes: 200, 406

Sample Response for 200:

```
{
  "num_loads": 0,
  "loads": []
}
```

Sample Request for 406:

Accept_mimetypes: text/html

Sample Response for 406:

```
{
  "Error": "Requests a content type not supported by the endpoint."
}
```

Endpoint: POST /loads

Protected: No

Possible return status codes: 201

Sample Request for 201:

```
{
  "weight": "1000kg",
  "creation_date": "2/10/2010",
  "length": "10m"
}
```

Sample Response for 201:

```
{
  "carrier": null,
  "weight": "1000kg",
  "creation_date": "2/10/2010",
  "length": "10m",
  "self": "https://final-project-314709.uc.r.appspot.com/loads/5667525748588544",
  "id": 5667525748588544
}
```

Endpoint: GET /boats/boat_id

Protected: Yes

Possible return status codes: 401, 200, 404, 403,

Sample Request for 200:

Header: Authorization = Bearer {{jwt1}}

Sample Response for 200:

```
{
  "name": "neo",
  "type": "a",
  "owner": 5634161670881280,
  "creation_date": "2/10/2019",
  "loads": [],
  "id": 5681556802764800
}
```

Endpoint: DELETE /boats/boat_id

Protected: Yes

Possible return status codes: 401, 404, 403, 204

Sample Request for 204:

Header: Authorization = Bearer {{jwt1}}

Endpoint: PATCH or PUT /boats/boat_id

Protected: Yes

Possible return status codes: 401, 404, 403, 400, 202

Sample Request for 202:

Header: Authorization = Bearer {{jwt1}}

```
{
  "type": "Nano",
  "creation_date": "2/10/2014"
}
```

Sample Response for 202:

```
{
  "owner": 5634161670881280,
  "name": "neo",
  "creation_date": "2/10/2014",
  "loads": null,
  "type": "Nano",
  "id": 5681556802764800,
  "self": "https://final-project-314709.uc.r.appspot.com/boats/5681556802764800"
}
```

Endpoint: GET /loads/load_id

Protected: No

Possible return status codes: 404, 406, 200

Sample Response for 200:

```
{
  "carrier": null,
  "weight": "1000kg",
  "length": "10m",
  "creation_date": "2/10/2010",
  "id": 5748214695198720
}
```

Endpoint: PUT/PATCH /loads/load_id

Protected: No

Possible return status codes: 404, 400, 202

Sample Request for 202:

```
{
  "weight": "10g",
  "creation_date": "7/12/2015"
}
```

Sample Response for 202:

```
{
  "carrier": null,
  "weight": "10g",
  "creation_date": "7/12/2015",
  "length": "10m"
}
```

Endpoint: DELETE /loads/load_id

Protected: No

Possible return status codes: 204

Endpoint: POST /boats/boat_id/loads/load_id

Protected: Yes

Possible return status codes: 401, 404, 403, 400, 204

Endpoint: DELETE /boats/boat_id/loads/load_id

Protected: Yes

Possible return status codes: 401, 404, 403, 204

Endpoint: GET /users

Protected: No

Possible return status codes: 200

Sample Response for 200:

```
[
  {
    "user_id": "auth0|60abdd6b959e8a0069ac5b4b",
    "id": 5634161670881280
  },
  {
    "user_id": "auth0|60abdd7ed2cc62006ab63b91",
    "id": 5644004762845184
  }
]
```